

ÉCOLE SUPÉRIEURE PRIVÉE D'INGÉNIERIE ET DE TECHNOLOGIE
ESPRIT

RAPPORT DE STAGE

Stage de Développement

Développement d'un Agent IA WhatsApp

Automatisation Intelligente de la Relation Client

Entreprise d'accueil	Établissement
Carthage Dev Agence Web & Solutions Digitales	ESPRIT École d'Ingénieurs - Tunisie

Stagiaire Mohamed Ali	Encadrant Professionnel M. Majdi Abbes
---------------------------------	--

Période : 01 Août 2025 - 31 Août 2025

Année Universitaire 2024-2025

Table des Matières

Remerciements

Je tiens à exprimer ma profonde gratitude à toute l'équipe de **Carthage Dev** pour leur accueil chaleureux, leur disponibilité et leur soutien tout au long de mon stage.

Je remercie tout particulièrement **M. Majdi Abbes**, mon encadrant professionnel, pour sa patience, son accompagnement et ses précieux conseils qui m'ont permis d'acquérir de nouvelles compétences et d'approfondir mes connaissances dans le développement d'agents intelligents et l'automatisation via **n8n** et **WhatsApp Web.js**.

Ce stage a été une expérience très enrichissante, tant sur le plan professionnel que personnel. Il m'a permis de découvrir le monde du développement d'applications d'intelligence artificielle en entreprise, de mettre en pratique mes connaissances et de renforcer ma motivation pour ce domaine passionnant.

Je suis reconnaissant à **Carthage Dev** de m'avoir offert cette opportunité qui a marqué une étape importante dans mon parcours de formation.

Introduction

Contexte du stage

Dans le cadre de ma formation à **ESPRIT**, j'ai effectué un stage de développement d'une durée d'un mois (du 01/08/2025 au 31/08/2025) au sein de l'entreprise **Carthage Dev**. Ce stage m'a offert l'opportunité de travailler sur un projet innovant combinant l'intelligence artificielle et la messagerie instantanée.

Objectifs du stage

Les objectifs principaux de ce stage étaient les suivants :

- Développer mes compétences en intégration d'API et en automatisation de processus
- Mettre en pratique les connaissances théoriques en intelligence artificielle et développement web
- Découvrir le monde professionnel et les méthodes de travail en entreprise
- Réaliser un projet concret répondant à un besoin réel du marché

Problématique

Comment automatiser intelligemment la gestion des conversations WhatsApp avec les clients en utilisant l'intelligence artificielle, tout en maintenant une qualité de service optimale et une expérience utilisateur fluide ?

Dans un contexte où la communication client devient de plus en plus exigeante, les entreprises cherchent des solutions pour répondre rapidement et efficacement aux demandes de leurs clients 24/7. L'utilisation d'un agent IA peut répondre à ce besoin tout en réduisant la charge de travail des équipes support.

Chapitre 1 : Présentation de l'Entreprise

1.1 Introduction

Carthage Dev est une entreprise tunisienne spécialisée dans le **développement de logiciels et les solutions numériques**. Elle offre une large gamme de services incluant la conception, la programmation, le conseil technique ainsi que l'accompagnement digital des entreprises et des particuliers.

1.2 Présentation générale de Carthage Dev

Carthage Dev est une **agence web complète** proposant des services variés tels que la création de sites web, le référencement (SEO), l'hébergement, la messagerie professionnelle, la conception d'identité visuelle et le web design. Son équipe jeune, dynamique et compétente travaille en étroite collaboration avec les clients à chaque étape du projet.

1.3 Structure organisationnelle




Carthage Dev est organisée autour de trois pôles principaux :

Pôle Design	Pôle Développement	Pôle Client
Création graphique, identité visuelle, logos et conception UI	Développement sur mesure, applications web et maintenance	Communication client, compréhension des besoins et suivi

Chapitre 2 : Description du Projet

2.1 Présentation générale

Le projet **AI WhatsApp Agent** consiste à créer un agent conversationnel intelligent capable de répondre automatiquement aux messages WhatsApp des clients, sans intervention humaine. Ce système combine trois technologies principales :

 ChatGPT Intelligence Artificielle	 n8n Orchestration Workflows	 WhatsApp-web.js Intégration WhatsApp
---	---	--

2.2 Objectifs du projet

Objectif principal

Développer un agent intelligent qui peut comprendre et répondre aux demandes des clients de manière naturelle et contextuelle.

Objectifs spécifiques

- Automatiser les réponses aux questions fréquentes
- Gérer des processus complexes comme les réservations
- Maintenir une conversation fluide et naturelle
- Réduire le temps de réponse aux clients
- Permettre un service client disponible 24/7

2.3 Cas d'usage : Agence de voyage

Pour illustrer les capacités du système, prenons l'exemple d'une **agence de voyage** proposant des réservations d'hôtels en Tunisie :

Étape	Description
1. Prise de contact	Le client envoie un message pour réserver un hôtel
2. Collecte d'infos	L'agent demande : dates, nombre de personnes, type d'hôtel
3. Proposition	L'agent propose une sélection d'hôtels avec tarifs
4. Confirmation	Confirmation de réservation et envoi des détails

Chapitre 3 : Analyse et Conception

3.1 Analyse des besoins

L'analyse des besoins a permis d'identifier les fonctionnalités essentielles et les contraintes techniques nécessaires à la mise en œuvre de l'agent conversationnel WhatsApp intelligent.

Besoins fonctionnels

- **Réception et traitement des messages** : écouter les nouveaux messages, identifier l'expéditeur et filtrer les contenus
- **Analyse sémantique** : comprendre le sens et l'intention de l'utilisateur
- **Gestion du contexte** : maintenir la continuité des échanges via Simple Memory
- **Mécanisme de fallback** : gérer les situations non comprises

Exigences non fonctionnelles

Critère	Valeur
Temps de réponse	< 3 secondes
Disponibilité	99,9%
Sécurité	Chiffrement E2E

3.2 Flux de traitement des messages

Le processus de traitement des messages suit un flux précis et structuré :

Étape 1 : Réception du message

- Le client envoie un message sur WhatsApp
- whatsapp-web.js capture le message
- Le bot filtre le message (vérifie qu'il n'est pas d'un groupe)
- Les données sont formatées : {text, from, number, timestamp}

Étape 2 : Envoi vers n8n

- Le bot envoie une requête POST au webhook n8n
- Transmission des données JSON
- Gestion du timeout (30 secondes max)

Étape 3 : Traitement dans n8n

- Webhook reçoit les données
- Vérification des règles prédéfinies (mots-clés, FAQ)
- Si règle détectée : Réponse directe, sinon envoi à l'AI Agent

Étape 4 : Traitement par l'AI Agent

- Récupération du contexte depuis Simple Memory
- Consultation de la base de connaissances (Google Sheets)
- Envoi à OpenAI avec prompt système et historique
- Génération de la réponse adaptée

Étape 5 : Post-traitement et réponse

- Vérification et formatage de la réponse (max 4000 caractères)
- Mise à jour de la mémoire et sauvegarde dans Google Sheets
- Envoi du message sur WhatsApp via msg.reply()

3.3 Architecture technique

Vue d'ensemble de l'architecture du système :

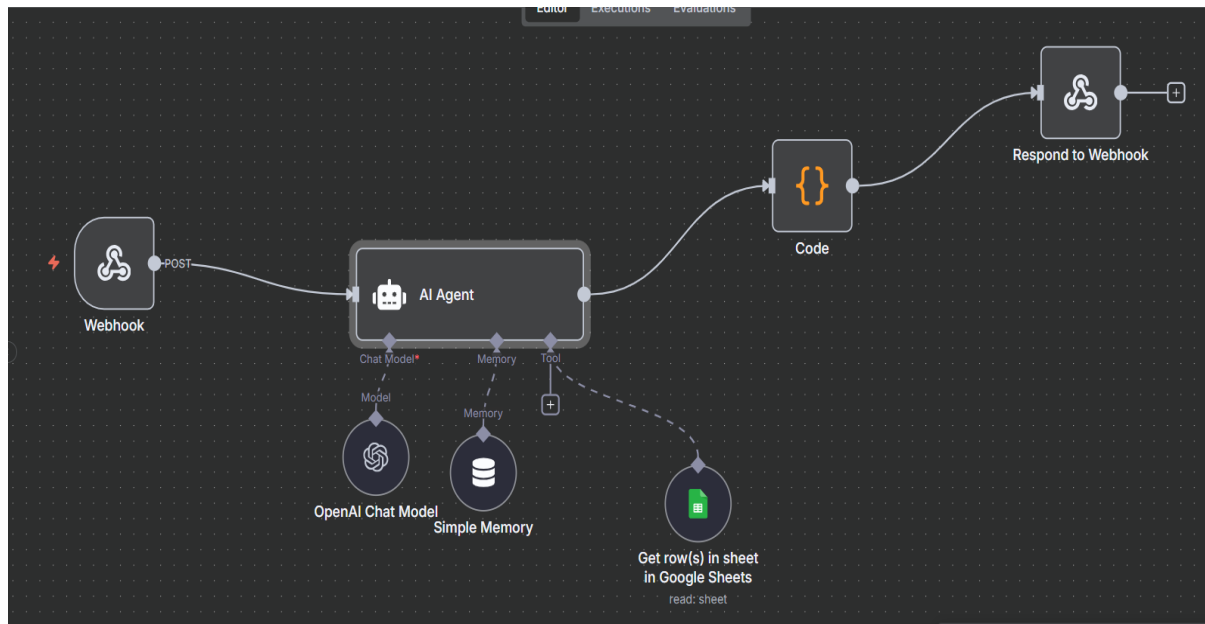


Figure 1 : Architecture globale du système

3.4 Composants principaux

Composant	Rôle et Fonctionnalités
Bot WhatsApp	Interface entre WhatsApp et n8n - Authentification QR, réception/envoi messages
Plateforme n8n	Orchestrateur central - Gestion flux, intégration OpenAI et Google Sheets
OpenAI ChatGPT	Moteur IA - Analyse sémantique, génération de réponses naturelles
Simple Memory	Gestion mémoire conversationnelle - Historique et contexte par utilisateur
Google Sheets	Base de données - Sauvegarde conversations, données clients, FAQ

3.5 Workflow n8n détaillé

Réception des données via Webhook

Le processus débute par un **nœud Webhook**, configuré pour recevoir les requêtes POST provenant du bot WhatsApp. Ce point d'entrée collecte les données JSON contenant le texte du message, le numéro de l'expéditeur et l'horodatage.

Interaction avec l'agent IA

Une fois les données reçues, elles sont transmises à un **nœud AI Agent**, qui gère la communication avec le modèle **OpenAI Chat Model**. Ce dernier s'appuie sur la mémoire conversationnelle fournie par **Simple Memory** et interagit avec **Google Sheets** pour consulter la base de connaissances.

Génération de la réponse avec OpenAI

Le modèle OpenAI utilisé (GPT-3.5-Turbo ou GPT-4) reçoit un prompt contenant l'historique de la conversation, les instructions système et le message actuel. Les paramètres comme la **température (0,7)** et le nombre maximal de **tokens (500)** assurent un équilibre entre créativité et précision.

Traitement et formatage des réponses

Les résultats produits sont transmis à un **nœud Code Node**, chargé d'appliquer la logique métier spécifique : validation et formatage du texte, extraction de données structurées, adaptation de la réponse à la longueur maximale (4 000 caractères).

Chapitre 4 : Développement et Réalisation

4.1 Environnement de développement

Outils et technologies

```
"whatsapp-web.js": "^1.23.0",
"axios": "^1.6.0",
"express": "^4.18.2",
"qrcode-terminal": "^0.12.0"
```

Figure 2 : Bibliothèques principales utilisées

Outil/Technologie	Utilisation
Visual Studio Code	IDE de développement
Node.js / JavaScript	Langage de programmation
Render.com	Plateforme de déploiement cloud
n8n (cloud)	Plateforme d'automatisation

4.2 Développement du Bot WhatsApp

Configuration initiale

```
const { Client, LocalAuth } = require('whatsapp-web.js');
const axios = require('axios');
const express = require('express');
const qrcode = require('qrcode-terminal');

const app = express();
app.use(express.json());

let isReady = false;

// URL du webhook n8n
const N8N_WEBHOOK_URL = 'https://n8n-04z5.onrender.com/webhook/whatsapp-in';
```

Figure 3 : Configuration du bot WhatsApp

La configuration initiale du bot WhatsApp repose sur l'utilisation de la bibliothèque **whatsapp-web.js**, qui permet de simuler la connexion à l'application WhatsApp Web à partir d'un environnement Node.js.

Authentification WhatsApp

```

  > const client = new Client({
  >   authStrategy: new LocalAuth({
  >     clientId: "whatsapp_bot"
  >   }),
  >   puppeteer: {
  >     headless: true,
  >     args: [
  >       '--no-sandbox',
  >       '--disable-setuid-sandbox',
  >       '--disable-dev-shm-usage',
  >       '--disable-gpu'
  >     ]
  >   }
  > });
  // Gestion du QR code
  > client.on('qr', qr => {
  >   console.log('Scannez le QR code pour vous connecter:');
  >   qrcode.generate(qr, { small: true });
  > });
  // Bot pret
  > client.on('ready', () => {
  >   isReady = true;
  >   console.log(' Bot pret à l\'utilisation!');
  > });

```

Figure 4 : Processus d'authentification QR Code

Captures d'écran du système

```

client.on('message', async msg => {
  // Filtrage
  if (msg.fromMe || msg.from.includes('@g.us') || msg.type !== 'chat')
    return;

  console.log(`Message de ${msg.from}: ${msg.body}`);

  try {
    // Préparation des données
    const requestData = {
      text: msg.body.trim(),
      from: msg.from,
      number: msg.from.replace('@c.us', ''),
      timestamp: new Date().toISOString()
    };

    // Envoi vers n8n
    const response = await axios.post(N8N_WEBHOOK_URL, requestData, {
      headers: { 'Content-Type': 'application/json' },
      timeout: 30000
    });

    // Extraction de la réponse
    let botReply = response.data.reply ||
      response.data.message ||
      response.data.text ||
      'Désolé, je n'ai pas pu traiter votre demande.';
  }
});

```

Figure 5 : Interface n8n

```

// Limitation de longueur
if (botReply.length > 4000) {
  botReply = botReply.substring(0, 3900) +
    '\n\n... (réponse raccourcie)';
}

// Envoi de la réponse
await msg.reply(botReply);
console.log(`Réponse envoyée`);

} catch (error) {
  console.error('Erreur:', error.message);
  await msg.reply('Désolé, une erreur s'est produite.');
```

Figure 6 : Conversation WhatsApp

4.3 Logique de filtrage des messages

Une phase de filtrage est mise en place afin d'optimiser le fonctionnement du bot :

- **msg.fromMe** : ignorer les messages envoyés par le bot lui-même
- **msg.from.includes('@g.us')** : exclure les messages des groupes WhatsApp
- **msg.type !== 'chat'** : filtrer les contenus non textuels

Chapitre 5 : Compétences Acquises

5.1 Compétences techniques

Domaine	Compétences
Backend Development	Node.js, Express.js, API REST
Intelligence Artificielle	OpenAI API, Prompt Engineering
Automatisation	n8n Workflows, Webhooks
Déploiement Cloud	Render.com, Gestion serveurs

5.2 Compétences professionnelles

- **Autonomie** : capacité à travailler de manière indépendante sur des tâches complexes
- **Rigueur** : méthodologie et organisation dans le travail
- **Communication** : échanges efficaces avec l'équipe et présentation des résultats
- **Adaptabilité** : apprentissage rapide de nouvelles technologies

Chapitre 6 : Journal des Tâches Réalisées

Semaine 1 (01/08 - 07/08)

1. **Jour 1** : Création du compte n8n, découverte de l'interface et des fonctionnalités
2. **Jour 2** : Étude détaillée de n8n, création d'un premier workflow de test
3. **Jour 3** : Test du nœud Webhook, tests avec Postman
4. **Jour 4** : Installation de whatsapp-web.js, étude de la documentation
5. **Jour 5** : Création serveur Express.js, première connexion WhatsApp réussie

Semaine 2 (08/08 - 14/08)

1. **Jour 6** : Configuration connexion serveur-webhook n8n
2. **Jour 7** : Test d'intégration WhatsApp-n8n, premier succès !
3. **Jour 8** : Création workflow réponses automatiques
4. **Jour 9** : Jour de repos, révision documentation
5. **Jour 10** : Intégration OpenAI, réponses IA fonctionnelles

Semaine 3 (15/08 - 21/08)

1. **Jour 11** : Ajout Simple Memory, conversations cohérentes
2. **Jour 12** : Intégration Google Sheets, sauvegarde automatique
3. **Jour 13** : Création scénarios de conversation types
4. **Jour 14** : Refactorisation du code, gestion d'erreurs
5. **Jour 15** : Système de logging détaillé

Semaine 4 (22/08 - 29/08)

1. **Jour 16** : Scénarios de secours (Fallback)
2. **Jour 17** : Rédaction guide utilisateur
3. **Jour 18** : Sauvegarde et backup des configurations
4. **Jour 19** : Simulation de cas réel
5. **Jour 20** : Démonstration interne devant l'équipe
6. **Jour 21** : Optimisation finale, projet finalisé et opérationnel

Conclusion

Bilan du stage

Ce stage d'un mois au sein de l'entreprise **Carthage Dev** a constitué une expérience formatrice et enrichissante. J'ai eu l'opportunité de concevoir et de développer un **agent conversationnel intelligent intégré à WhatsApp**, combinant plusieurs technologies modernes telles que **Node.js**, **n8n**, **OpenAI** et **Google Sheets**.

Apports personnels

Sur le plan **technique**, j'ai découvert et maîtrisé l'automatisation avec n8n, l'intégration de bibliothèques complexes et l'utilisation d'APIs d'intelligence artificielle. J'ai aussi renforcé mes bases en développement backend et en debugging.

Sur le plan **professionnel**, cette immersion m'a permis de mieux comprendre le fonctionnement du milieu de l'entreprise : respect des délais, travail en équipe et adaptation aux contraintes de production.

Perspectives d'évolution

Plusieurs pistes d'amélioration ont été identifiées pour la suite du projet :

Évolutions à court terme

- Création d'un dashboard d'administration pour le suivi en temps réel
- Amélioration de l'interface de configuration des réponses
- Optimisation des performances et réduction de la latence

Évolutions à moyen terme

- Extension multi-plateforme (Telegram, Messenger, Discord)
- Support multilingue pour une clientèle internationale
- Reconnaissance d'images et traitement de fichiers multimédias
- Intégration de paiements en ligne

Évolutions à long terme

- Intégration de machine learning pour l'amélioration continue
- Outils analytiques avancés et tableaux de bord BI
- Intégration avec des solutions métiers (CRM, ERP, systèmes de réservation)
- Développement d'une version white-label pour revente

Réflexion personnelle

Ce stage a représenté bien plus qu'une simple mission technique. Il m'a permis de **confirmer mon orientation professionnelle** vers le développement d'applications intelligentes, combinant backend, IA et automatisation.

J'y ai découvert mon goût pour la résolution de problèmes complexes, la conception de solutions concrètes et l'apprentissage continu. Cette expérience m'a également permis d'identifier mes **forces** (curiosité, méthode, persévérance) ainsi que mes **axes d'amélioration**, notamment en architecture logicielle, en tests automatisés et en communication technique en anglais.

Ce stage a été une expérience riche et déterminante dans mon parcours. Je ressors de cette expérience avec une vision claire de mon avenir professionnel, une méthodologie solide, et une motivation profonde pour continuer à innover dans le domaine du développement logiciel et de l'intelligence artificielle.

Le Stagiaire

Mohamed Ali

L'Encadrant

M. Majdi Abbès